SECURE SOFTWARE ARCHITECTURE APPROACH FOR ROLE-BASED ACCESS CONTROL USING ASPECT-ORIENTED DESIGN

MUNEER ABDULLAH SAEED HAZAA

FSKTM 2010 11

SECURE SOFTWARE ARCHITECTURE APPROACH FOR ROLE-BASED ACCESS CONTROL USING ASPECT-ORIENTED DESIGN

By

MUNEER ABDULLAH SAEED HAZAA

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfillment of the Requirements for the Degree of Doctor of Philosophy

September 2010

Dedicated to my beloved family:

My parents, Abdullah , karmah

Dedicated to my wife,

To my kids, Suhaib, Mays, and Roba;

To my family.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the Degree of Doctor of Philosophy

SECURE SOFTWARE ARCHITECTURE APPROACH FOR ROLE-BASED ACCESS CONTROL USING ASPECT-ORIENTED DESIGN

By

MUNEER ABDULLAH SAEED HAZAA

September 2010

Chairman **Abdul Azim Abd Ghani, PhD**

Faculty : Computer Science and Information Technology

Organizations define and enforce AC policies to protect sensitive information resources. The policy imposes requirements to ensure that only authorized users have access to the sensitive information resources. Normally, systems for various applications operate with different access control requirements. Currently, there exist different AC models to fulfill different requirements, such as mandatory access control (MAC) model, discretionary access control (DAC) model, the Chinese Wall model, and Role-based Access Control (RBAC) model. Consequently, a general AC service means that it supports multiple AC models, hence satisfying different applications.

Moreover, access control presents itself as a crosscutting concern, that is, it spans multiple object-oriented classes. However, implementing the access control requirements with the conventional object-oriented technique does not fully fulfil the modularization of crosscutting functionality. Because of different access control requirements, access control services should be flexible and extensible.

This thesis proposes a framework for role-based access control mechanism for RBAC using an aspect-oriented technique at architectural level. An aspect-oriented technique provides the explicit means to modularize crosscutting concerns in modularity units called aspects. Aspect-oriented technique could encapsulate the access control services as crosscutting concerns. RBAC is selected as the model since it is a well accepted AC model. Instead of individually implementing the mechanism supporting individual AC models, a more general AC service can be designed by supporting the RBAC model only. Thus, the framework provides flexibility in designing a secure system using role-based access control (RBAC) model. Moreover, an aspect-based role-based access control framework for CORBA authentication services has also been developed and formally verified. Two case studies have been implemented to verify the workability and the security properties of the proposed framework.

In the case studies, the core RBAC mechanism in the framework was organized in an object-oriented design, while each extension was captured as an aspect. This has resulted in a flexible and modularized framework that supports modularization of crosscutting functionality. This framework can be easily extended to fit any new access control requirements.

The thesis uses the Predicate/Transition Net (PrTN) to formally verify security properties of the proposed framework. The formal specification written in PrTN was translated into Promela, and verified using SPIN model checker. The security properties of the case studies were correct as expressed in temporal logic formulas.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

SENI BINA PERISIAN SELAMAT UNTUK KAWALAN CAPAIAN BERASASKAN PERANAN DENGAN MENGGUNAKAN REKA BENTUK BERORIENTASI ASPEK

Oleh

MUNEER ABDULLAH SAEED HAZAA

September 2010

Pengerusi: Abdul Azim Abd Ghani, PhD

Fakulti: Sains Komputer dan Teknologi Maklumat

Organisasi mentakrif dan menguatkuasa polisi kawalan capaian (KC) untuk melindungi sumber maklumat sensitif. Polisi tersebut mengenakan keperluan untuk menjamin hanya pengguna berautoriti mempunyai capaian ke sumber maklumat sensitif. Kebiasaannya, sistem untuk pelbagai aplikasi beroperasi dengan keperluan kawalan capaian yang berbeza. Pada masa ini terdapat model KC yang berlainan untuk memenuhi keperluan yang berbeza seperti model Kawalan Capaian Mandatori (KCM), model Kawalan Capaian Budibicara (KCB), model *Chinese Wall*, dan model Kawalan Capaian Berasaskan Peranan (KCBP). Akibatnya, perkhidmatan KC yang umum bermakna ianya menyokong banyak model KC, dengan itu memenuhi aplikasi yang berlainan.

Tambahan pula, kawalan capaian merupakan urusan potong-memotong, iaitu ia menjangkau banyak kelas berorientasi-objek. Walau bagaimanapun, mengimplemen keperluan kawalan capaian dengan teknik berorientasikan objek konvensional tidak

dapat secara sepenuhnya memenuhi pemodulan kefungsian potong-memotong. Disebabkan oleh keperluan untuk kawalan capaian yang berlainan, perkhidmatan kawalan capaian seharusnya fleksibel dan boleh diperluas.

Tesis ini mencadangkan satu rangka untuk mekanisma kawalan capaian berasaskan peranan untuk KCBP dengan menggunakan teknik berorientasi-aspek di peringkat seni bina. Teknik berorientasi-aspek menyediakan cara eksplisit untuk memodulkan urusan potong-memotong dalam unit bermodul yang dipanggil aspek. Teknik berorientasi-aspek dapat mengurung perkhidmatan kawalan capaian sebagai urusan potong-memotong. KCBP dipilih sebagai model kerana ianya adalah model KC yang sudah diterima baik. Daripada mengimplemen mekanisma suatu model KC secara individu, lebih baik perkhidmatan KC yang lebih umum direka bentuk dengan hanya menyokong model KCBP. Oleh itu, rangka tersebut menyediakan kefleksibelan dalam mereka bentuk sistem selamat dengan menggunakan model kawalan capaian berasaskan peranan (KCBP). Tambahan pula, rangka kawalan capaian berasaskan peranan untuk perkhidmatan pengesahan CORBA juga dibina dan disahkan secara formal. Dua kajian kes telah diimplemen untuk mengesah kebolehkerjaan dan sifat keselamatan rangka tang dicadangkan.

Dalam kajian kes tersebut, mekanisma teras KCBP dalam rangka tersebut dibentuk dalam reka bentuk berorientasi-objek, sementara setiap perluasan digambarkan sebagai aspek. Ini telah menghasilkan rangka yang fleksibel dan bermodul yang menyokong pemodulan fungsian potong-memotong. Rangka ini adalah senang untuk diperluas bagi memenuhi sebarang keperluan kawalan capaian baharu.

Tesis ini menggunakan *Predicate/Transition Net (PrTN)* untuk mengesah secara formal sifat keselamatan rangka yang dicadangkan. Spesifikasi formal tertulis dalam *PrTN* diterjemah ke *Promela,* dan disahkan dengan menggunakan model penyemak SPIN. Sifat keselamatan kajian kes adalah betul seperti terungkap dalam formula logik temporal.

ACKNOWLEDGEMENTS

In the name of ALLAH, the Beneficent and the Compassionate. Praise to ALLAH for giving me strength, patience, and motivation to complete this research work. I would like to take this opportunity to record my gratitude towards the peoples who have given me support during the phases of this research. My deepest appreciation and gratitude go to the research committee leads by Professor Dr. Abdul Azim Abd Ghani, who has always taken time to listen to my ideas. He has patiently answered my questions, provided invaluable guidance, fruitful discussion, and encouragement.

I also would like to record my great thanks to all members of my Ph.D. supervisory committee; Associate Prof. Dr. Ali Bin Mamat and Associate Prof. Dr. Hamidah Ibrahim for their support, attentions during my research work and the guidance in each discussion during all steps of this work and. They has helped me more than I expected for providing me inspiration for this work and also for her virtuous guidance, encouragement and help during the time of doing the research.

A great thank to the Faculty of Computer Science and Information Technology, the university library and Universiti Putra Malaysia that provided the working environment for performing this work.

I would also like to thank the Dean Secretary, Puan Norhaidah and the faculty Deputy Dean Secretary.

My thanks are also extended to my friends and colleagues, for sharing experiences throughout the years.

MUNEER ABDULLAH SAEED HAZAEA

September 2010

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	Access Control
AO	Aspect-Oriented
AOD	Aspect Oriented Design
AOM	Aspect-Oriented Modeling
AOP	Aspect-Oriented Programming
AOSD	Aspect–Oriented Software Design
CCC	Crosscutting Concern
CORBA	Common Object Request Broker Architecture
CORBASec	CORBA Security Services
CSP	Constraint Satisfaction Problem
CTL	Computation Tree Logic
DAC	Discretionary Access Control
DSD	Dynamic Separation Of Duty
GRM	Granted Rights Matrix
HRBAC	Hierarchy Role Base Access Control
LTL	Linear Temporal Logic
MAC	Mandatory Access Control
OBS	Predefined Data Sets
OO	Object-Oriented
OOP	Object-Oriented Programming
OPS	Objects Operations
ORB	Object Request Broker
PA	Permission Assignment
PRMS	Permissions
PROMELA	Process Meta Language
PrTN	Predicate-Transition Net
RBAC	Role-Based Access Control
RH	Role Hierarchy
UA	User Assignment
UML	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Access Control (AC) decision making process is a function, which ensures protection of a resource through granting or denying permission to access the resource. Nowadays, information systems are increasingly interconnected and accessible in a network environment (Samarati & De Capitani di Vimercati, 2001; Vimercati, 2006). As a result, security issues like confidentiality, integrity and availability are of foremost concern in the modern information enterprise. Having reliable AC service to protect distributed resources has become a challenge for any organization. For instance, factors like size of an enterprise; the subjects and objects involved in security polices; and the special requirements of different applications have contributed to the complexity of having a reliable AC services. It is therefore important to develop an extensible, well-modularized, customizable and reliable role-based access control (RBAC) service to alleviate some of the complexities in the design stage of any secure system.

Separation of concerns is a general principle in software engineering introduced to control the complexity of ever-growing programs. Aspect-oriented programming (AOP) has been proposed to improve separation of concerns in software to adapt with any software extensions. AOP is based on the idea that computer systems are better programmed by separately specifying the various concerns of a system and their

relationships, and then relying on the mechanisms in the underlying AOP environment to weave or compose them together into a coherent program.

The confluence of AOP and software architecture research was pointed out in (Georg, Ray, & France, 2002), and security concerns such as authentication and access control, were suggested to be placed in architectural connectors.

In this work, an AOP for designing secure software architectures is proposed. A secure software architecture defines the structure of the software system and the interaction and coordination among its components, which correctly enforces the security requirement. An aspect-oriented software architecture design consists of a model of essential functionality and a number of aspect models, each modularized around a specific concern. We focus on the aspects that reflect security concerns, which is are mainly RBAC concerns. The integrated model of the system is obtained by "weaving" the model of essential functionality with the aspect models.

1.2 Problem Statement

The AC mechanism is often implemented as a part of operating systems, database management systems or middleware systems. These systems work with various applications with different access control requirements. Therefore, the AC mechanism should be suitable to the system concerned. An AC mechanism should support at least one AC model to perform its operation. In fact, the access requirements from different organizations - military, governmental and business are not the same. As a result, there

exist different AC models, such as mandatory access control (MAC) (Graham, 1972; Harrison, Ruzzo, & Ullman, 1976) discretionary access control (DAC) model (Bell & LaPadula, 1973; Biba, 1977; Denning, 1976; LaPadula & Bell, 1996) and the Chinese wall model (Brewer & Nash, 1989). Consequently, a general AC service means that it supports multiple AC models, hence satisfying different applications.

The RBAC model is another widely accepted AC model, besides MAC and DAC models. Currently, RBAC does not support the extensible, customizable and modularized design. With the advantage of being policy neutral, which means it can simulate other AC models, through proper configuration that is instead of individually implementing the mechanism supporting each AC model, a general AC service can simulate the implementation processes of different AC models by supporting RBAC model. This results in simplifying the development process. Moreover, designers will focus on how to implement an RBAC service as a general AC model.

The implementation of a RBAC service is a complicated and tedious process task (Samarati & De Capitani di Vimercati, 2001), so it is not wise to implement each RBAC sub-model from zero. As Parnas indicated (Parnas, 1978); (Kiczales et al., 2001), the software designer should be aware that he is not designing a single program but a family of programs. By extensibility, it demands that the AC service keeps up with the evolution of technology and new RBAC variants can be easily supported. By customizability, it requires the AC service can be configured to support any RBAC sub-model. By modularity, it actually represents a collection of design requirements including comprehensibility, maintainability, and reusability. For example if *RBACy* is

based on $RBAC_X$, then the implementation of $RBAC_Y$ should be able to reuse the implementation of $RBAC_X$.

Since the conventional object-oriented programming languages do not have appropriate linguistic means and mechanisms to facilitate unplanned and non-invasive adaptation, so some requirements such as extensibility, customizability and modularity are difficult to be satisfied at the same time (Czarnecki et. al, 2000). In other words, object-oriented languages do not support modularization of crosscutting functionality. It is also apparent that, these features have to be implemented at the structural level using advanced design techniques such as Aspect-Oriented Design (AOD) techniques. Furthermore, an AC service also requires high assurance. An authorized access should be granted, and any unauthorized access must be denied. To find a potential defect, a common practice is to test an AC implementation extensively. But it is impractical to test all data and all paths, due to a combinatorial explosion (Collofello, 1988). So testing cannot guarantee that an AC implementation is bug free. In addition, errors detected at the end of the development are more costly to rectify. In contrast, formal verification can prove the correctness of a system using mathematical methods. Formal verification can be introduced early in the phase of development. In fact, at the design phase, formal verification can be used to verify the correctness of an AC design.

However, a formal verification is impossible without a formal specification (Wing, 1990). To ensure the correctness of an AC implementation, it is necessary to formally specify the design of the AC implementation first. This thesis is concerned with implementing of more general AC services by supporting RBAC services making AC

services extensible and customizable. By adopting the techniques of the AOD such difficulties can be resolved at the software architecture level.

In brief, the following requirements are faced by AC service designers:

- Different AC models should be supported in a more general Access Control service and that can be achieved via proper configuration of the RBAC model.
- To support different AC models, the designed services should be extensible, customizable, and well modularized.
- To provide high assurance, an AC design must be formally specified so that the design can be formally analysed.

1.3 Objectives of the Research

The objective of this research is to propose an approach that leads to the design of well-modularized, extensible, customizable and reliable role-based access control. Complexities arising from the need to support applications with different RBAC requirements can be handled at the design stage. The main objectives are:

(i) To propose an Aspect Oriented Programming–Role Base Access Control framework for the RBAC at the architectural level, so that it can be easily extended to fit new access control requirements.

(ii) To develop a Role Based Access Control framework using CORBA authentication services based on aspect design.

1.4 Scope of the Research

This research focuses on a particular area of AC security, namely, RBAC security, and its suitability to be re-implemented using AOP attached to a general model, that can adopt different AC models. In other words, it is customable and extendable enough to be used by different AC models.

CORBA authentication services are used since CORBA security interfaces provide access control that separates the security concerns from the application concerns. This separation makes it clear to be designed using AOD. In these interfaces, a domain is a distinct scope over which one security policy is enforced. Each domain has one and only one domain access policy object. There may be sub-domains for different aspects of a policy. Each sub-domain also has a domain access policy object.

Formal analysis for an aspect-oriented design can help to determine whether two aspects are orthogonal or not, and whether an aspect's quantification is correctly defined. But that would be beyond the scope of this thesis. Although this research addresses the problem dynamically, it does not mean that the modification should occur at run time. It means that the modification in the aspect-oriented model can be done in a much easier way than that in the object oriented (OO) model. The advantage of this research is that

the additional code for implementing new functionalities can always be kept in a separate location rather than having it spread across the existing code like the OO model on its own.

1.5 The Research Approach

Given the above one work mentioned complexities and requirements, it is essential to design an approach that implements RBAC services in a systematic way. The proposed approach (Figure 1.1) combines both aspect-oriented design model and formal methods.

In this approach, base models are expressed in the Unified Modeling Language (UML), where the Class diagrams specify static structure and sequence diagrams describe how objects collaborate (interact) to accomplish tasks and the sequence diagram presents a behavioral view that focuses on the interactions that take place between class objects when they collaborate to accomplish a specific task. Aspect security models are identified according to the UML class parameterized elements, such as, relationships and classes that consist of attributes and operations.

Formal analysis and verification follow to check the reliability of the approach the AOP to be employed in this research, allows the researcher to dynamically modify the static object oriented (OO) model, thus creating a system that grows to meet new requirements. The aspect-oriented design (AOD) is a natural extension of AOP. The success of AOP in many applications has motivated the researcher to apply it to the design of extensible AC services.

The proposed approach works at the design level has two benefits. First, it will achieve the high assurance. The approach gives a process to formally model the AOD and formal verification is applied to detect potential defects. Second, the product of the proposed approach is an aspect-oriented design framework, the implementation of which is not limited to any specific programming language. Developers may select the implementation language based on their expertise and the customer's requirements. The Predicate/Transition Nets (PrTN) comprises formal notations, selected to build formal design models. The main components of the approach are: first, identifying and analyzing the security points that can be driven from the UML class diagrams and the weaving interfaces with the security mode. Second, the formal analyses which involve the translating of the aspect models and then to PrTN. Third, Checker models are applied for verifying the design.

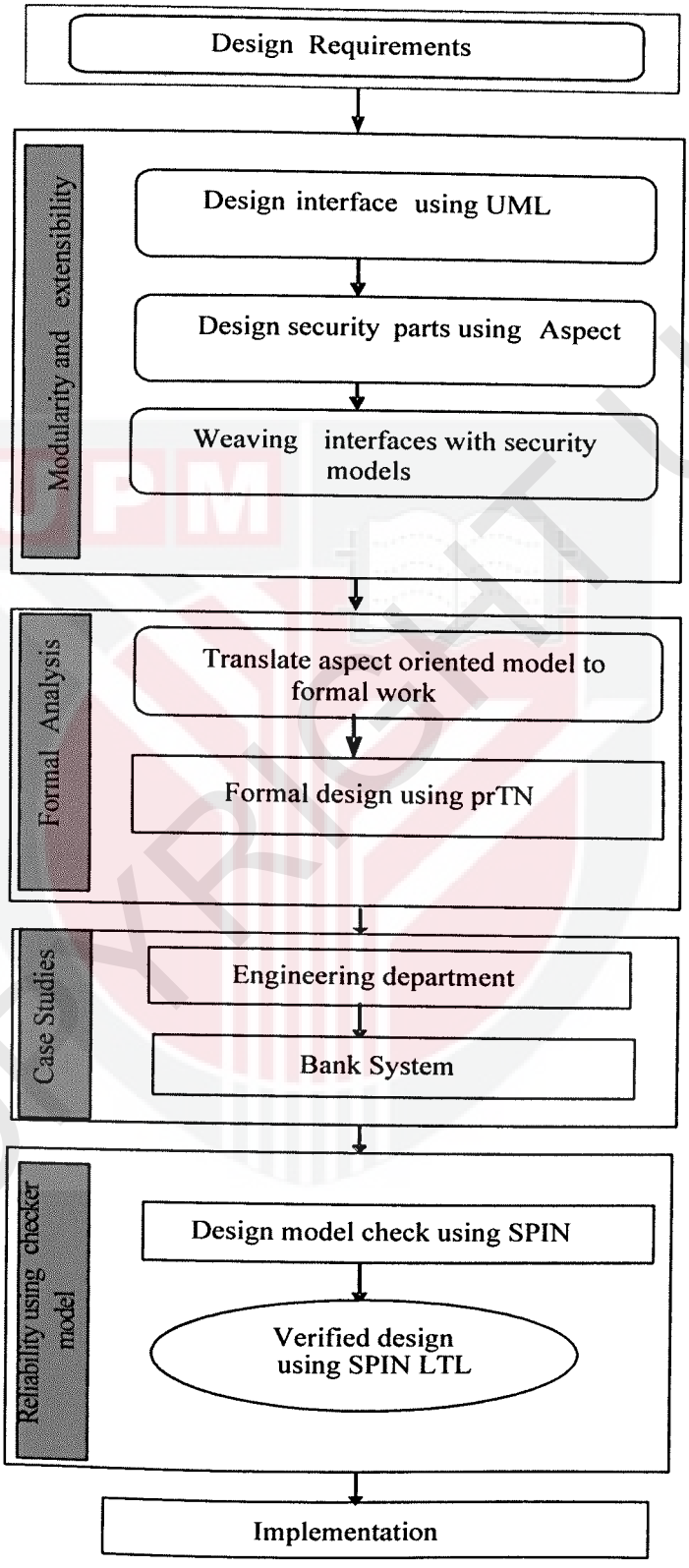

Figure 1.1. Research Approach

1.6 Structure of Thesis

This thesis is organized into six chapters. Chapter 1 gives the background information of current access control made operational in systems to protect distributed resources; problems encountered and the objectives of this study specifically to address some problematic issues using the object oriented approach. Chapter 2 introduces and explains the main concepts that was introduced in Chapter1 as in addition to the related works and reviews of concerns underlying this study. Chapter 3 describes the research methodology and its general architecture, which was adopted to develop a security approach for an aspect-oriented design of an AC framework. Within this design, the design guidelines and principles in the proposed approach are also illustrated. Chapter 4 presents and describes the designed model of the aspect-oriented specification notations and modelling process of the RBAC. Chapter 5 provides the main results of the study and discusses the process of the formal analysis. In chapter 6, we have the conclusions and the recommendations for future work.

REFERENCES

Aldawud, O., Elrad, T., & Bader, A. (2003). *UML profile for aspect-oriented software development*. In Proc. *Third International Workshop Modeling,* Aspect Oriented modeling.

Anderson, R. J. (2001). *Security Engineering: A guide to building dependable distributed systems*: John Wiley & Sons, Inc. New York, NY, USA.

Andrews, J. H. (2001). Process-algebraic foundations of aspect-oriented programming. *In Proc. Third International Conference on Metaleuel Architectures and Separation of Crosscutting Concerns (2001), Lecture (Reef Nleoctteison in Computer voSlcuimenec e, 2192, p1ag87e-s209. Springer-Verlag.*

Barker, S., & Stuckey, P. J. (2003). Flexible access control policy specification with constraint logic programming. *ACM Transactions on Information and System Security, 6*(4), 501-546.

Basch, M., & Sanchez, A. (2003). *Incorporating Aspects into the UML*. In *Proc. Third International Workshop on aspect oriented modeling.*

Basin, D., Doser, J., & Lodderstedt, T. (2006). Model driven security: From UML models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology, 15*(1), 39-91.

Bell, D. E., & LaPadula, L. J. (1973). *Secure computer systems: Mathematical foundations*: Technical Report MTR-2547.

Biba, K. J. (1977). Integrity considerations for secure computer systems: Storming Media. Mitre TR-3153, Mitre Corporation.

Blaze, M., Feigenbaum, J., & Lacy, J. (1996). *Decentralized trust management.* In *Proc .IEEE Symposium Security and Privacy.*

Boner, J., (2005). *Aspect werkz 2 and the road to AspectJ 5.* Paper presented at the invited Talk at AOSD 2005, Industry track.

Brand, S. L. (1985). DoD 5200.28-STD Department of Defense Trusted Computer System Evaluation Criteria (Orange Book). *National Computer Security Center.*

Brewer, D. F. C., & Nash, M. J. (1989). *The Chinese wall security policy. . In Proc. IEEE Symposium on Security and Privacy,. In Proc. IEEE Symposium on Security and Privacy,* pages 206-214, Oakland, California.

Brucker, A. D., Doser, J., & Wolff, B. (2006). A model transformation semantics and analysis methodology for SecureUML. In Model Driven Engineering Languages and Systems 9th *International Conference, Models 2006,* volume 4199 of Lecture Notes in *Computer Science, pages 306-320, Geneva, Italy, 2006. Springer Berlin.*

Bryant, R. E. (1986). Graph-based algorithms for boolean function manipulation. *IEEE Transactions on computers, 35*(8), 677-691.

Chandramouli, R. (1999). *Implementation of multiple access control policies within a CORBASEC framework. In Proc 22nd National Information Systems Security Conference,* Crystal City, Virginia, USA.

Cheng, B. H. C., Konrad, S., Campbell, L. A., & Wassermann, R. (2003). *Using security patterns to model and analyze security requirements. IEEE Workshop on Requirements for High Assurance Systems* (RHAS03).

Chitchyan, R., Rashid, A., & Sawyer, P. (2005). *Comparing requirements engineering approaches for handling crosscutting concerns.* In *Workshop on Requirements Engineering* (held with CAiSE), Porto, Portugal.

Clarke, S. (2002). Extending standard UML with model composition semantics. *Science of Computer Programming, 44*(1), 71-100.

Clarke, S., & Baniassad, E. (2005). Aspect Oriented analysis and design. Addison-Wesley Professional, Boston.

Clarke, S., & Walker, R. J. (2002). *Towards a standard design language for AOSD.* In Proc. AOSD 2002, Enschede, The Netherlands.

Clarke, S., Harrison, W., Ossher, H., & Tarr, P. (1999). *Subject-oriented design: towards improved alignment of requirements, design, and code. In Proc. OOPSLA.*

Collofello, J. S. (1988). Introduction to software verification and validation: *CARNEGIE -MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.*

Cooper, K., Dai, L., & Deng, Y. (2005). Performance modeling and analysis of software architectures: An aspect-oriented UML based approach. *Science of Computer Programming, 57*(1), 89-108.

Courtois, P. J. (1985). On time and space decomposition of complex structures. *Communications of the ACM (CACM),* 28(6):590-603.

Czarnecki, K., Eisenecker, U., Gluck, R., Vandevoorde, D., & Veldhuizen, T. (2000). Generative programming and active libraries. *Lecture notes in computer science*, 25-39.

David, F. F., Dennis, M. G., & Nickilyn, L. (1993). *An examination of federal and commercial access control policy needs.*

De Win, B., Piessens, F., & Joosen, W. (2006). *How secure is AOP and what can we do about it? , SESS'06. Shanghai, China. Copyright 2006 ACM 1-59593-085-X/06/0005.*

Demir, Ö., Dévanbu, P., Wohlstadter, E., & Tai, S. (2007). *An aspect-oriented approach to bypassing middleware layers.*

Denning, D. E. (1976). A lattice model of secure information flow. *Communications of the ACM*, 19(5):236.

Dijkstra, E. W., & Dijkstra, E. W. (1976). *A discipline of programming*: prentice-hall Englewood Cliffs, NJ.

Evans, A., France, R., & Grant, E. (1999). *Towards formal reasoning with uml models.* In: *Proceedings of the OOPSLA'99 Workshop on Behavioral Semantics.*

Ferraiolo, D. F., Barkley, J. F., & Kuhn, D. R. (1999). A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security, 2*(1), 34-64.

Ferraiolo, D. F., Kuhn, D. R., Chandramouli, R., & Barkley, J. (2006). Role Based Access Control (RBAC). Web page, March 2006.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC), 4(3), 224-274.*

Ferraiolo, D., Kuhn, R., & Sandhu, R. (2007). RBAC standard rationale: Comments on "A critique of the ANSI standard on role-based access control". Security & Privacy. *IEEE, 5*(6), 51-53.

Ferraiolo, M. D. F., Gilbert, M. D. M., & Lynch, M. N. (1995). *An examination of federal and commercial access control policy needs.* In *Proc. 15th National Computer Security Conference.*

Filman, R. E., & Friedman, D. P. (2000). *Aspect-oriented programming is quantification and obliviousness. In Workshop on Advanced Separation of Concerns,* OOPSLA 2000, Minneapolis, MN, USA.

Filman, R. E., Elrad, T., Clarke, S., & Ak it, M. (2005). Aspect-Oriented Software Development. : *Addison-Wesley.* ISBN 0-32121-976-7.

France, R., Ray, I., Georg, G., & & Ghosh, S. (2004). An aspect-oriented approach to early design modeling. *IEE Proceedings-Software, 151(4), 173-185.*

Furnell, S. M., Jusoh, A., & Katsabas, D. (2006). The challenges of understanding and using security: A survey of end-users. *Computers & Security, 25*(1), 27-35.

Gao, D. (2006). *Aspect-Oriented Middleware.* Citeseer. *PhD thesis. University of Toronto.*

Genrich, H. J., & Lautenbach, K. (1981). System modeling with high-level Petri nets.

THEORET. COMP. SCI., 13(1), 109-136.

Georg, G., France, R., & Ray, I. (2002). *An aspect-based approach to modeling security concerns. Proceedings of the Workshop on Critical Systems Development with UML (CSDUML '02),* held in conjunction with *the 5th International Conference* on the Unified Modeling Language (UML '02), pages 107-120.

Georg, G., Houmb, S. H., & Ray, I. (2006). Aspect-Oriented Risk Driven Development of Secure Applications. *Lecture Notes in Computer Science, 4127*, 282.

Georg, G., Ray, I., & France, R. (2002). *Using aspects to design a secure system. ", Proceedings of the 8th IEEE International Conference on Engineering of Complex Computer Systems* (ICECCS '02), pages 117-126.

Giuri, L. (1999). *Role-based access control on the web using Java.* In *Proc. 4th ACM Workshop on Role-Based Access Control*, Fairfax, VA, USA.

Goldberg, A. (1984). *SMALLTALK-80: the interactive programming environment*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Gosling, J., Joy, B., & Guy Jr, L. (1996). Steele. The Java Language Specification: Addison Wesley. ISBN 0-201-63451-1.

Gradecki, J. D., Lesiecki, N., & Gradecki, J. (2003). *Mastering AspectJ: aspect-oriented programming in Java*: *Wiley Publishing.*

Grady Booch, J. R., and Ivar Jacobson . . (2001). *The Unified Modeling Language User Guide.* . Addison-Wesley.

Graham, G. K. (2005). *Protection - principles and practice*: Personal correspondence. *Proc. SJCC*, AFIPS, 40, 1972.G. Kiczales. Personal correspondence.

Graham, G. S., & Denning, P. J. (1972). *Protection--principles and practice. Proc. SJCC, AFIPS, 40, 1972.G. Kiczales.*

Hanenberg, S. (2006). Design Dimensions of Aspect-Oriented Systems. PhD thesis, Duisburg Essen University, Essen, Germany.

Hanenberg, S., Stein, D., & Unland, R. (2005). Roles from an aspect-oriented perspective. In VAR'05: *Views, Aspects and Roles Workshop,* ECOOP 2005, Glasgow, UK.

Harrison, M. A., Ruzzo, W. L., & Ullman, J. D. (1976). Protection in operating systems. *Communications of the ACM*, 19(8):461-471.

Hashii, B. (2004). *Lessons learned using alloy to formally specify MLS-PCA trusted security architecture.* In Proc. *FMSE Workshop association the 10twh ith ACM Conference* on Computaenrd Communications 2004.

He, X. (2001). PZ nets—a formal method integrating Petri nets with Z. *Information and Software Technology, 43*(1), 1-18.

He, X., & Lee, J. A. N. (1990). Integrating predicate transition nets with first order temporal logic in the specification and verification of concurrent systems. *Formal Aspects of computing, 2*(1), 226-246.

Hirschfeld, R. (2003). Aspects-aspect-oriented programming with squeak. In NetObjectDays '02: Proceedings of *International Conference NetObjectDays - Objects,Components, Architectures, Services, and Applications for a Networked World*,volume 2591 of Lecture Notes in Computer Science, pages 216-232, Erfurt, Germany . Springer-Verlag. ISBN 3-540-00737-7.

Holzmann, G. J. (2004). *The SPIN model checker: Primer and reference manual*: Addison-Wesley Professional.

Hussmann, H., Demuth, B., & Finger, F. (2002). Modular architecture for a toolset supporting OCL. *Science of Computer Programming, 44*(1), 51-69.

Jackson, A., Sánchez, P., Fuentes, L., & Clarke, S. (2006). *Towards traceability between ao architecture and ao design.* . In EA '06: *In Workshop of Early Aspects at AOSD 2006,* Bonn, Germany.

Jackson, D. (2001). Micromodels of software: Modelling and analysis with Alloy. *MIT Lab for Computer Science.*

Jacobson, I. (2003). Case for Aspects-Part I. Software Development Magazine: pages 32-37, October 2003. Pages 42-48.

Jajodia, S., Samarati, P., Sapino, M. L., & Subrahmanian, V. S. (2001). Flexible support for multiple access control policies. *ACM Transactions on Database Systems, 26*(2), 214-260.

Jensen, K. (1997). *Coloured Petri nets: basic concepts, analysis methods and practical use*: Basic Concepts. EATCS Monographs on Theoretical Computer Science. Springer-Verlag.

Jonscher, D., & Dittrich, K. R. (1995). Argos-A configurable access control system for interoperable environments. *Database Security, IX: Status and Prospects*, 43-60.

J.D.Gradecki,N.Lesiecki. *Mastering AspectJ.* Wiley, 2003.

Jürjens, J. (2005). *Secure systems development with UML*: Springer Verlag. Springer, Berlin Heidelberg, New York.

Kan, C. Y., & He, X. (1995). High-level algebraic Petri nets. *Information and Software Technology, 37*(1), 23-30.

Karjoth, G. (2000). Authorization in CORBA security. *Journal of Computer Security, 8*(2), 89-108.

Katara, M., & Katz, S. (2003). *Architectural views of aspects*. In *Proc. AOSD* 2003, Boston, MA, USA.

Kiczales, G. J., Lamping, J. O., Lopes, C. V., Hugunin, J. J., Hilsdale, E. A., & Boyapati, C. (2002). Aspect-oriented programming: In ECOOP '97: *Proceedings of European Conference on Object-Oriented Programming*, volume 1241 of LNCS, pages 220-242, Jyv'askyl'a, Finland. Springer-Verlag.

Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., & Griswold, W. (2001). Getting started with AspectJ. *Communications of the ACM (CACM)*, 44(10):59-65, 2001a.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C. V., Loingtier, J. M., et al. (1997). Aspect-Oriented Programming, In proceedings of the European Conference on Object-Oriented Programming (ECOOP), Finland: Springer-Verlag LNCS.

Koch, M., & Parisi-Presicce, F. (2003). *Formal access control analysis in the software development process*. In Proc. *FMSE Workshop in association with the 9th ACM Conference* on Computer and Communications Security.

Krechetov, I., Tekinerdogan, B., Garcia, A., Chavez, C., & Kulesza, U. (2006). *Towards an integrated aspect-oriented modeling approach for software architecture design*. In *8th International Workshop on Aspect-Oriented Modeling*, AOSD '06, Bonn, Germany.

Krishnamurthi, S., Fisler, K., & Greenberg, M. (2004). *Verifying aspect advice modularly. In Proc. SIGSOFT'Od/FSE-12, Newport Beach, CA, USA.*

Kugblenu, F. M., & Asim, M. (2007). *Separation of Duty in Role Based Access Control System: A Case Study. In Proc. SIGSOFT'Od/FSE-12, Newport Beach, CA, USA.*

LaPadula, L. J., & Bell, D. E. (1996). Secure computer systems: A mathematical model. *Journal of Computer Security, 4*, 239-263.

Loopez-Grao, J. P., Merseguer, J., & Campos, J. (2004). From UML activity diagrams to Stochastic Petri nets: application to software performance engineering. *Proceedings of the Fourth International Workshop on Software and Performance (WOSP'04), pp. 25 – 36.*.

Laddad, R. *AspectJ In Action* . MANNING, 2003.

Lorenz, D. H. K., S. (2006). *Feature interaction in AspectJ/5.* Paper presented at the Conference Proceedings.

Mandl, K. D., Simons, W. W., Crawford, W. C. R., & Abbett, J. M. (2007). Indivo: a personally controlled health record for health information exchange and communication. *BMC Medical Informatics and Decision Making, 7*(1), 25.

Massoni, T., Gheyi, R., & Borba, P. (2004). A UML class diagram analyzer. *In: Third Workshop* on Critical Systems Development with UML, UML 2004, and Lisbon, Portugal 100-114.

Moser, M., Ibens, O., Letz, R., Steinbach, J., Goller, C., Schumann, J., et al. (1997). SETHEO and e-SETHEO-the CADE-13 systems. *Journal of Automated Reasoning, 18*(2), 237-246.

Neumann, G., & Strembeck, M. (2001). *Design and implementation of a flexible RBAC-service in an object-oriented scripting language*. In *Proc. ACM Conference on Computer and Communications Security (CCS)*, Philadelphia, PA, USA.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM, 15*(12), 1058.

Parnas, D. L. (1976). On the design and development of program families.. *IEEE Transactions on Software Engineering*, SE-2(1), 1-9.

Parnas, D. L. (1978). *Designing software for ease of extension and contraction.*

Pawlak, R., Duchien, L., Florin, G., Legond-Aubry, F., Seinturier, L., & Martelli, L. (2002). *AUML notation for aspect-oriented software design.* . Paper presented at the In Aspect-Oriented Modeling with UML Workshop at AOSD, Enschede, and the Netherlands.

Paz-Trillo, C., & Rocha, V. (2005). Architectural Patterns to Secure Applications with an Aspect Oriented Approach.

Rinard, M., Salcianu, A., & Bugrara, S. (2004). A classification system and analysis for aspect-oriented programs. *In Aspect-Oriented Modeling with UML Workshop at AOSD 2002*, Enschede, the Netherlands.

Rushby, J. (1995). *Formal methods and their role in the certification of critical systems.* Citeseer. Technical Report CSL.

Röck, A., & Kresman, R. (2003). *On Petri Nets and Predicate-Transition Nets.*

Samarati, P., & De Capitani di Vimercati, S. (2001). Access control: Policies, models, and mechanisms. *Lecture notes in computer science*, 137-196.

Sandhu, R. S. (1998). Role-based Access Control1. *The Engineering of Large Systems, 46*, 237.

Sereni, D., & de Moor, O. (2003). *Static analysis of aspects*. In Proc. *International*

Conference on Aspect-Oriented Software Developmeent.

Smith, R. (2005). Introduction to multilevel security. *Handbook of Information Security.*

Spinczyk, O., Gal, A., & Schröder-Preikschat, W. (2002). *AspectC++: An Aspect-Oriented Extension to the C++ Programming Language*. In CRPIT '02: Proceedings *of the Fortieth Inter national Conference* on Tools Pasic, pages 53{60, Sydney, *Australia. Australian Computer Society, Inc. ISBN 0-909925-88-7.*

Stallings, B. W. (2007). *Role-Based Access Control in Computer Security*

Stein, D., Hanenberg, S., & Unland, R. (2004). Query models. In UML '04: *Proceedings of the 7th International Conference on the Unified Modeling Language,* volume 3273 of LNCS, pages 98-112, Lisbon, Portugal. Springer-Verlag.

Stein, D., Hanenberg, S., & Unland, R. (2006). *Expressing different conceptual models of join point selections in aspect-oriented design*. In *AOSD '06: Proceedings of the 5th International Conference on Aspect-Oriented Software Development*, pages 15-26, Bonn, Germany, ACM Press.

Stepney, S., & Lord, S. P. (1987). Formal specification of an access control system. *Software: Practice and Experience, 17*(9), 575-593.

Stroustrup, B. (1997). *The C++ programming language*: Addison-Wesley, third edition, ISBN 0-201-32755-4.

Sutton Jr, S. M., & Rouvellou, I. (2002). *Modeling of software concerns in Cosmos*. In Proc.*International Conference on Aspect*

Sutton Jr, S. M., & Rouvellou, I. (2005). Concern modeling for aspect-oriented software development. *Aspect-Oriented Software Development*, 479–505.

Tari, Z., & Chan, S. W. (1997). A role-based access control for intranet security. *IEEE Internet Computing, 1*(5), 24-34.

Team, A. J. (2006). The AspectJ project at Eclipse.org. (Vol., [April 16th], (2006).

Trillo, C. P., & Rocha, V. (2005). *Architectural patterns to secure applications with an aspect-oriented approach. Proceedings of the 5th Latin American Conference* on Pattern Language of Programming, page 89-105.

Wand, M., Kiczales, G., & Dutchyn, C. (2004). A semantics for advice and dynamic join points in aspect-oriented programming. *ACM Transactions on Programming Languages and Systems, 26*(5), 890-910.

Whittle, J., & Araujo, J. (2004). Scenario modeling with aspects. *IEE Proceedings-Software,* Vol151, Issue 4, pages 157-171.

Viega, J., Bloch, J. T., & Chandra, P. (2001). Applying aspect-oriented programming to security. *Cutter IT Journal, 14*(2), 31-39.

Vimercati, d. P., S. D. C.& Samarati, S.P.(2006). Access control: principles and solutions. *Handbook of Information Security*, 406.

Wing, J. M. (1990). A specifier's introduction to formal methods. 8-10. *IEEE Computer*, 23(9):8-10.